

ϵ - Optimal Stopping Time for Genetic Algorithms

C. A. Murthy, Dinabandhu Bhandari and Sankar K. Pal*

*Machine Intelligence Unit
Indian Statistical Institute
203 B. T. Road
Calcutta 700 035, India
sankar@isical.ac.in*

Abstract. In this article, the concept of ϵ -optimal stopping time of a genetic algorithm with elitist model (EGA) has been introduced. The probability of performing mutation plays an important role in the computation of the ϵ -optimal stopping times. Two approaches, namely, pessimistic and optimistic have been considered here to find out the ϵ -optimal stopping time. It has been found that the total number of strings to be searched in the optimistic approach to obtain ϵ -optimal string is less than the number of all possible strings for sufficiently large string length. This observation validates the use of genetic algorithms in solving complex optimization problems.

1. Introduction

Genetic algorithms (GAs) are stochastic search methods based on the principles of natural genetic systems [9, 12, 14]. They perform a multidimensional search in providing an optimal solution for evaluation (fitness) function of an optimization problem. Unlike the conventional search methods, GAs deal simultaneously with multiple solutions and use only the fitness function values. GAs are theoretically and empirically found to provide global near optimal solutions

* Address for correspondence: Machine Intelligence Unit, Indian Statistical Institute, 203 B. T. Road, Calcutta 700 035, India

of various complex optimization problems in the fields of operations research, VLSI design, pattern recognition, image processing, machine learning etc. [1, 2, 3, 4, 7, 16, 17, 18, 26].

While solving an optimization problem using GAs, each solution is coded as a string (called "chromosome") of finite length (say, L). Each string or chromosome is considered as an individual. A collection of M (M is finite) such individuals is called a population. GAs start with a randomly generated population of size M . In each iteration, a new population of same size is generated from the current population using three basic operations on the individuals of the population. The operators are (i) Reproduction/Selection, (ii) Crossover and (iii) Mutation.

The new population obtained after selection, crossover and mutation is then used to generate another population. Note that the number of possible populations is always finite since M is finite. Usually, GAs run for a fixed number of iterations or terminate if no improvement is found for a fixed number of iterations. No stopping criterion, which ensures the optimality, is available in the literature to the best of our knowledge. In GAs, the knowledge about the best string obtained so far is preserved either within the population or in a separate location outside the population; in that way the algorithm would report the best value found, among all possible coded solutions obtained during the whole process. This paper deals with the GAs with the elitist model (EGA) of selection of De Jong [29], where the best string obtained in the previous iteration is copied into the current population if the fitness function values of all strings are less than the previous best.

Some theoretical aspects of genetic algorithms have been studied in the literature. The results mainly rely on string representation of solutions and on the notion of a schema [12, 14]. Schema is a template for string and it allows the exploration of similarities among strings. According to the schema theorem of Holland, the selection process increases (exponentially) the sampling rates of the average schemata. But it does not guarantee the convergence of the process to the global optimal solution. Vose extended this work and interpreted GAs as constrained random walk and generalized the concept of schemata [23]. There are other articles also on the analysis of schema distribution and deceptive problems [8, 13, 19].

Genetic algorithms have been successfully modelled as Markov chains [5, 10, 20, 24, 28]. Vose [24], and Davis et. al. [10] have not preserved the knowledge of the previous best in their model. Bhandari et. al. [5], Rudolph [20], Suzuki [27] preserved the knowledge of the previous best in their model and proved the convergence of GAs to the optimal string. To the best of our knowledge the optimal stopping time for GAs has not been provided in the literature.

In this paper, ϵ -optimal stopping criterion for GAs with elitist model of De Jong [14] has been provided and consequently ϵ -optimal stopping time has been derived. The properties of ϵ -optimal stopping time have been studied. Optimal mutation probability has been provided in the context of the worst case analysis. Subsequently, ϵ -optimal stopping time has been derived for GAs with elitist model of De Jong under an "optimistic" assumption.

The basic principles of genetic algorithms and a description of GAs with elitist model (EGAs) are provided in the next section. The mathematical modelling of EGAs and the convergence are discussed in sections 3. Section 4 and 5 deal with the optimal stopping time of EGAs.

2. Basic Principles of Genetic Algorithms

We describe the basic principle of GAs in this section, considering a problem of maximizing a function $f(x)$, $x \in D$ where D is a finite set. The problem here is to find x_{opt} such that

$$f(x_{opt}) \geq f(x); \forall x \in D$$

Note that D is a discrete domain since D is finite.

2.1. Chromosomal Representation and Population

A string of length L over a finite set of alphabet $\mathcal{A} = \{\alpha_1, \alpha_2, \dots, \alpha_a\}$ is used as a chromosomal representation of the solution x . Each string S corresponds to a value x in D and is of the form

$$S = (\beta_L \beta_{L-1} \dots \beta_2 \beta_1); \beta_i \in \mathcal{A}, \forall i$$

For example, if we consider $\mathcal{A} = \{0, 1\}$ then a binary string of length L is a chromosomal or string representation of a solution. Henceforth, throughout this article, we shall take $\mathcal{A} = \{0, 1\}$.

Note that the total number of strings (i.e., the number of different values for the variable x) is 2^L . GAs find the optimal string among these 2^L strings. A set of M such strings forms a population for an iteration. To generate an initial population, generally, a random sample of size M is drawn from 2^L strings. ♣

2.2. Genetic operators

This subsection discusses the genetic operators while generating a new population in an iteration. In every iteration, we evaluate each chromosome of the population using fitness function fit . Evaluation or fitness function fit for a string S is equivalent to the function f :

$$fit(S) = f(x),$$

where S corresponds to x . Without loss of generality, we are assuming here that $fit(S) > 0$ for all S in \mathcal{S} and the optimization problem under consideration is a maximization problem. If f takes negative values, then a suitably large value is added to every $f(x)$ to make $fit(S)$ greater than zero for every S .

A mating pool is generated using selection and the chromosomes of the mating pool undergo crossover and mutation. Moreover, it is customary to preserve the knowledge of the best chromosome obtained so far during the process.

Selection is an artificial version of natural selection, a Darwinian survival of the fittest among string creatures. Usually, in this process, individual strings of the current population are copied into a mating pool with respect to the empirical probability distribution based on their fitness function values. In this process some chromosomes would be selected more than once and the chromosomes with low fitness values would not be selected.

Crossover exchanges information between two potential strings and generates two offspring for the next population. $M/2$ pairs are made randomly from the population. Then the crossover is performed with probability p (crossover probability). In this paper, we are going to deal with the single point crossover operation.

Mutation : Mutation is an occasional random alteration of a character position. Mutation introduces some extra variability into the population. Though it is performed usually with very low probability q (> 0), it has an important role in the generation process. Every character β_i , $i = 1, 2, \dots, L$ in each chromosome (generated after crossover) has an equal chance to undergo mutation. Note that, any string can be generated from any given string by mutation operation. This result is stated in the form of the following lemma.

Lemma 1. *Probability of generating any string S_1 from a given string S_2 is greater than zero and its value is $q^\nu(1 - q)^{L-\nu}$ where ν ($0 \leq \nu \leq L$) is the number of places where those two strings have distinct characters. (ν is the Hamming distance between S_1 and S_2).*

Proof:

Trivial. □

The mutation probability q is taken to be in the range of $(0, 0.5]$. It may be due to the fact that, intuitively, the probability of mutating i bit positions is more than that of mutating $i + 1$ bit positions, i.e.,

$$q^i(1 - q)^{L-i} \geq q^{i+1}(1 - q)^{L-i-1} \quad \forall i = 0, 1, 2, \dots, L - 1,$$

which results in $q \leq 0.5$. Hence the minimum probability of obtaining any string from any given string is q^L , that is, mutation needs to be performed at every character position of the given string.

The knowledge about the best string obtained so far can be preserved either (i) in a separate location outside the population or (ii) within the population. For case (i), a string (S_{opt}) is maintained, in which the best string is stored. Before performing the genetic operations the best string (S_{cur}) of the current population is compared with S_{opt} , and S_{opt} is replaced with S_{cur} if $fit(S_{opt}) < fit(S_{cur})$. We will call a GA with this strategy of retaining the knowledge of the best string obtained so far as *general genetic algorithm* (GGA). On the other hand, in case (ii), the best string of the current population is copied in a temporary location before performing the genetic operations and is recopied into the new population obtained at the end of the iteration. A way of implementing this strategy may be to replace the worst with the previous best when the fitness values of all individuals in the new population is less than or (equal to) the previous best. We refer to the GAs with this strategy as genetic algorithms with elitist model or EGAs. The other ways of implementing (ii) are : replace the worst of the present population with the previous best when the fitness value of the worst is less than the previous best or replace the worst offspring of the new population with the previous best. This paper deals with the genetic algorithms with elitist model or EGAs.

The basic steps in an EGA are described below.

EGA :

1. Generate an initial population Q of size M and calculate fitness value of each string S of Q .
2. Find the best string S_{cur} of Q .
3. Do the selection operation of Q to generate a mating pool Q_1 . Perform crossover and mutation operations on the strings of the mating pool and obtain a population Q_{tmp} .
4. Compare the fitness value of each string S of Q_{tmp} with S_{cur} . Replace the string having the minimum fitness value of Q_{tmp} with S_{cur} if the fitness value of each string of Q_{tmp} is less than $fit(S_{cur})$; otherwise no replacement takes place in Q_{tmp} . Rename Q_{tmp} as Q .
5. Go to step 2.

Note : Steps 2, 3 and 4 together make an iteration.

2.3. Genetic parameters

Genetic operations have been discussed, in detail, in the previous subsection. Note that the values for the parameters L, M, p and q have to be chosen 'properly' before performing those operations. Regarding M , it has already been stated that, M is taken as an even integer. The probability (p) of performing crossover operation is taken to be any value between 0.0 and 1.0. Usually in GAs, p is assumed to be a value in the interval $[0.25, 1]$. The mutation probability q is taken to be in the interval $(0, 0.5]$ whereas, usually, it lies in $[0.001, 0.01]$ [9]. It is also assumed in this paper that the crossover and mutation probabilities (p and q) remain fixed during the process.

Various genetic operations and their implementations have been described above. In an iteration, these operations are performed on a population and they result in a new population. In the next iteration, the new population will be subjected to these operations and the process goes on. It is to be noted here that, to the best of our knowledge, there exists no stopping criterion in the literature which ensures the convergence of GAs to an optimal solution. Usually, two stopping criteria are used in genetic algorithms. In the first, the process is executed for a fixed number of iterations and the best string, obtained so far, is taken to be the optimal one. While in the other, the algorithm is terminated if no further improvement in the fitness value for the best string is observed for a fixed number of iterations.

It has been stated that proportional selection strategy and single point crossover are components of the EGA mentioned here. However, it may be noted that the results in this article hold good for any selection strategy and for any crossover operation.

3. Convergence of Genetic Algorithms

3.1. Markov chain model of genetic algorithms

Researchers have successfully modelled genetic algorithms as Markov chains. The states of the chain are denoted by the populations of the genetic algorithms. A population Q is a collection

of strings of length L generated over a finite alphabet \mathcal{A} and is defined as follows :

$$Q = \{S_1, S_1, \dots, (\sigma_1 \text{ times}), S_2, S_2, \dots, (\sigma_2 \text{ times}), \dots, S_\xi, S_\xi, \dots, (\sigma_\xi \text{ times}) : S_i \in \mathcal{S}; \\ \sigma_i \geq 1 \text{ for } i = 1, 2, \dots, \xi; S_{r_1} \neq S_{r_2} \forall r_1 \neq r_2 \text{ and } \sum_{i=1}^{\xi} \sigma_i = M\}.$$

Let \mathcal{Q} denote the set of all populations of size M . The number of populations (N) or states in this Markov chain is finite.

In [10], T. E. Davis and C. J. Principe also modelled a GA as a Markov chain. The transition probabilities are calculated for several cases in their article. The theory for the existence of the stationary distribution for the transition from one population to another has been provided in [10], when the mutation probability is assumed to be a positive (> 0) constant. They also provide the sufficient condition, analogous with the simulated annealing, on the mutation probability parameter sequence to ensure that the nonstationary algorithm achieves limiting distribution.

Bhandari et. al. [5] proved the convergence the EGA to the optimal solution. The fitness function is taken into account in their model and they preserved the best string during the process. A summary of their work is given below for convenience. The concepts provided by them have been utilized in the derivation of ϵ -optimal stopping time. Note that other authors have also proved the convergence of EGA to the optimal string [20, 27]. The approach provided in this article for deriving stopping time is a direct consequence of the work of Bhandari et. al. [5].

3.2. Convergence of GAs

Genetic algorithms search over a space \mathcal{S} of 2^L strings and eventually provide the best with respect to the fitness function fit . The strings can be classified into a set of s classes depending on their fitness function values. The classes are defined as

$$\mathcal{S}_i = \{S : fit(S) = F_i\}$$

where F_i denotes the i th highest fitness function value. Thus $F_1 > F_2 > \dots > F_s$. Let us also assume without loss of generality that $F_s > 0$.

The fitness function value $fit(Q)$ of a population is defined as $fit(Q) = \max_{S \in Q} fit(S)$. Then the populations are partitioned into s sets. $E_i = \{Q : Q \in \mathcal{Q} \text{ and } fit(Q) = F_i\}$ is a set of populations having the same fitness function value F_i . In an iteration or generation, the genetic operators (selection, crossover and mutation) create a population $Q_{kl} \in E_k; l = 1, 2, \dots, e_k$ and $k = 1, 2, \dots, s$; from some $Q_{ij} \in E_i$ where e_k is the number of elements in E_k . The generation of a population Q_{kl} from Q_{ij} is considered as a transition from Q_{ij} to Q_{kl} and let $p_{ij.kl}$ denote this transition probability. Then the probability of transition from Q_{ij} to any population in E_k can be calculated as

$$p_{ij.k} = \sum_{l=1}^{e_k} p_{ij.kl}; j = 1, 2, \dots, e_i; i, k = 1, 2, \dots, s.$$

For all $j = 1, 2, \dots, e_i$ and $i = 1, 2, \dots, s$ one obtains

$$\begin{aligned} p_{ij.k} &> 0 && \text{if } k \leq i \\ &= 0 && \text{otherwise} \end{aligned} \quad (1)$$

by construction. This means that once GAs reach a population $Q \in E_k$ they will always be in some population $Q' \in E_k$ for $k \leq i$. In particular, once GAs reach a population $Q \in E_1$ they will never go out of E_1 .

Let $p_{ij.kl}^{(n)}$ be the probability that GA results in Q_{kl} at the n th step given that the initial state is Q_{ij} . Let $p_{ij.k}$ denote the probability of reaching one of the populations in E_k from Q_{ij} at the n th step. Then $p_{ij.k}^{(n)} = \sum_{l=1}^{e_k} p_{ij.kl}^{(n)}$.

To show the eventual convergence of a GA with elitist model to a global optimal solution the following theorem has been proved.

Theorem 1. For an EGA with the probability of mutation $q \in (0, \frac{1}{2}]$,

$$\lim_{n \rightarrow \infty} p_{ij.k}^{(n)} = 0 \text{ for } 2 \leq k \leq s; \forall j = 1, 2, \dots, e_i \text{ and } i = 1, 2, \dots, s.$$

$$\text{Hence } \lim_{n \rightarrow \infty} p_{ij.1}^{(n)} = 1 \quad \forall j = 1, 2, \dots, e_i \text{ and } i = 1, 2, \dots, s.$$

Proof:

It can be easily shown that

$p_{ij.1} > 0$ for $j = 1, 2, \dots, e_i$ and $i = 1, 2, \dots, s$ (from lemma 1). Let

$$\max_{i,j} (1 - p_{ij.1}) = \delta. \quad (2)$$

Note that $\delta < 1$ since $\min_{i,j} p_{ij.1} > 0$. Now,

$$\sum_{k \neq 1} p_{ij.k}^{(1)} = \sum_{k=2}^s p_{ij.k} = 1 - p_{ij.1} \leq \delta; \quad (3)$$

$$\begin{aligned} \sum_{k \neq 1} p_{ij.k}^{(2)} &= \sum_{k=2}^s \sum_{i_1 \neq 1}^{e_1} \sum_{j_1=1}^{e_{i_1}} p_{ij.i_1 j_1} p_{i_1 j_1.k} \quad (\text{since, } p_{1 j_1.k} = 0 \text{ for } k > 1) \\ &= \sum_{i_1 \neq 1}^{e_1} \sum_{j_1=1}^{e_{i_1}} p_{ij.i_1 j_1} \sum_{k=2}^s p_{i_1 j_1.k} \\ &= \sum_{i_1 \neq 1}^{e_1} \sum_{j_1=1}^{e_{i_1}} p_{ij.i_1 j_1} (1 - p_{i_1 j_1.1}) \\ &\leq \delta \sum_{i_1 \neq 1}^{e_1} \sum_{j_1=1}^{e_{i_1}} p_{ij.i_1 j_1} \quad (\text{from (2)}) \\ &= \delta \sum_{i_1 \neq 1} p_{ij.i_1} \\ &= \delta(1 - p_{ij.1}) \leq \delta^2; \end{aligned} \quad (4)$$

similarly, by mathematical induction, it can be shown that $\sum_{k \neq 1} p_{ij.k}^{(n)} \leq \delta^n$ for all ij .

Note that $\delta^n \rightarrow 0$ as $n \rightarrow \infty$ since $0 \leq \delta < 1$. Hence $\sum_{k \neq 1} p_{ij.k}^{(n)} \rightarrow 0$ as $n \rightarrow \infty$. Which, immediately implies $\lim_{n \rightarrow \infty} p_{ij.k}^{(n)} = 0$ for $2 \leq k \leq s$ for all i and j .

It is clear that,

$$\begin{aligned} \lim_{n \rightarrow \infty} p_{ij.1}^{(n)} &= \lim_{n \rightarrow \infty} (1 - \sum_{k \neq 1} p_{ij.k}^{(n)}) \\ &= 1. \quad \spadesuit \end{aligned}$$

Following conclusions can be made from the above Theorem :

- The proof is independent of the selection operation. The result is valid if any other selection operation is used, as long as the number of selected chromosomes is M .
- The proof is independent of the crossover operation too. The result holds good if any other crossover is used, as long as the number of strings generated after crossover is M .
- For any state (or population) Q_{ij} ; $i \geq 2$, $p_{ij.k}^{(n)} \rightarrow 0$ as $n \rightarrow \infty$, $\forall k \geq 2$. In other words, for sufficiently large number of iterations Q_{ij} will result in Q_{1l} for some $l = 1, 2, \dots, e_1$, i.e., the convergence to optimal string is assured with any initial population.
- The proof does not utilize the condition on q namely $q \leq 0.5$. Even if $0.5 < q < 1.0$, the proof is still valid. $q \leq 0.5$ will be utilized while discussing the stopping times. \clubsuit

4. Optimal Stopping Time

4.1. Foundation

In the previous section, the proof for convergence of EGA has been provided. According to theorem 1, an EGA will result in the best string as the number of iterations n goes to ∞ . But the process needs to be stopped for some finite value of n . The value of n at which the process is stopped is called the stopping time of the process. The objective here is to determine the value of n which is in some sense 'optimal'. More specifically, the problem is to find a value for n which will provide the best solution.

Note that, for a starting population Q , GAs can result in many populations with different probabilities after n iterations. Thus, if the process is to be stopped after a specific number of iterations, it may not always be guaranteed that the optimal string is obtained. In fact, the lemma stated below indicates that, there is a positive (> 0) probability of not obtaining the optimal solution after a finite number of iterations.

Lemma 2. *Let the fitness function be such that there exists a string S_0 which is not an optimal string (there always exists such an S_0 of the function is not constant). Let us consider the population Q_{ij} such that Q_{ij} contains M copies of S_0 . Then, the probability $p_{ij.ij}^{(n)}$ that the process will remain in Q_{ij} after n steps is positive (> 0) for all n .*

Proof:

Trivial. □

The above lemma provides a trivial result from the point of view of probability. But, it follows from the above lemma that no finite stopping time can guarantee the optimal solution. On the other hand, the process is to be terminated after finitely many iterations with the expectation that the process has achieved the optimal solution. Thus any decision regarding the stopping time should necessarily be probabilistic, since GA is a stochastic process.

The literature on stopping time for stochastic processes is vast [21]. We are stating the existing definition for ϵ -optimal stopping time below in the context of genetic algorithms.

Let $Q_{ij} \in E_i$ and $Q_{ij}^{(n)}$ denotes the population that is obtained at the end of the n th iteration of the genetic algorithm with the starting population as Q_{ij} . Let $g_{ij}^{(n)}$ denote the fitness function value of the population $Q_{ij}^{(n)}$. Let $\mathcal{E}(g_{ij}^{(n)})$ denote the expected value of $g_{ij}^{(n)}$, then

$$\mathcal{E}(g_{ij}^{(n)}) = \sum_{k=1}^i F_k p_{ij.k}^{(n)} \quad (5)$$

Now the ϵ -optimal stopping time of GA can be defined as follows.

Definition 1. Let $\epsilon \geq 0$ and N_0 be a positive integer. Then N_0 is said to be an ϵ -optimal stopping time for a GA if

$$n \geq N_0 \implies \mathcal{E}(g_{ij}^{(n)}) \geq F_1 - \epsilon \quad \forall i \text{ and } j. \spadesuit$$

In particular, if $\epsilon = 0$, N_0 is called 0-optimal stopping time or simply optimal stopping time [21]. Note that the above definition can not be used directly for GAs, since we do not know the value of F_1 . Thus we have modified the above definition slightly. From now onwards, the following definition will be referred as the ϵ -optimal stopping time for a GA.

Definition 2. (ϵ -optimal stopping time) : Let $\epsilon \geq 0$. Let N be a positive integer. Then N is said to be an ϵ -optimal stopping time for a GA if

$$n \geq N \implies \mathcal{E}(g_{ij}^{(n)}) \geq F_1(1 - \epsilon) \quad \forall i \text{ and } j. \spadesuit \quad (6)$$

Note that F_1 is used in the above definition too. But, the following manipulations remove the dependency of F_1 on the ϵ -optimal stopping time.

$$\mathcal{E}(g_{ij}^{(n)}) = \sum_{k=1}^i F_k p_{ij.k}^{(n)} \geq F_1 p_{ij.1}^{(n)} \quad \forall i \text{ and } j. \quad (7)$$

Thus, from (6) and (7), N needs to be found such that

$$n \geq N \implies p_{ij.1}^{(n)} \geq 1 - \epsilon \quad \forall i \text{ and } j. \quad (8)$$

Note that, if N is an ϵ -optimal stopping time for a GA then any $N_1 > N$ is also an ϵ -optimal stopping time for the GA. Thus for a given $\epsilon \geq 0$, we have to find N_0 such that N_0 is an ϵ -optimal stopping time for GA and $N < N_0$ would imply N is not an ϵ -optimal stopping time for the GA. In other words, we want to find the minimal ϵ -optimal stopping time for GA to reduce the computations.

Definition 3. (*Minimal ϵ -optimal time*) For an $\epsilon \geq 0$, N_0 is said to be minimal ϵ -optimal time of a GA if N_0 is an ϵ -optimal stopping time and $N < N_0$ would imply N is not an ϵ -optimal stopping time for the GA. ♠

Definition 4. (*ϵ -optimal string*) The best string obtained at the end of N iterations is called ϵ -optimal string where N is an ϵ -optimal stopping time for the GA ♠

Note also that ϵ can not take the value zero for a finite ϵ -optimal stopping time (lemma 2). Thus, from now onwards, ϵ is taken to be strictly greater than zero.

We have $p_{ij.1}^{(n)} > 0$ and $\lim_{n \rightarrow \infty} p_{ij.1}^{(n)} = 1$ (from theorem 1). This implies that for a given $\epsilon (> 0)$ there exists an N such that $p_{ij.1}^{(n)} \geq 1 - \epsilon$ for $n \geq N$. Hence, for a given $\epsilon > 0$, finite ϵ -optimal stopping time exists for a GA.

Now, the problem is to find the values for $p_{ij.1}^{(n)}$ for any initial population Q_{ij} . In other words, in solving a problem using GAs, once the values for $p_{ij.1}^{(n)}$ are known, the ϵ -optimal stopping time N can be determined. Since, the transition probabilities depend on the characteristics of the fitness function [5], we shall briefly discuss the fitness function below.

4.2. Fitness function

Extensive literature is available on the characteristics of the fitness function to expedite the search using GAs [12, 14]. The convergence depends also on the number of points or strings at which the maximum value occurs. It can be intuitively seen that if the number of strings having the maximum fitness function value is larger, the chance of fast convergence is higher.

Theorem 2. Let f_1 be a fitness function, which assumes s distinct values F_1, F_2, \dots, F_s , defined on S . Let S_1 and S_2 be two different strings such that $f_1(S_1) = f_1(S_2)$ and $f_1(S) \leq f_1(S_1) \forall S \in S$. Let us now define f_2 on S as follows :

$$f_2(S) = f_1(S), \quad \forall S \neq S_2 \text{ and } f_2(S_2) = F_i, \quad \text{for some } i > 1.$$

Let p^1 and p^2 be the transition probabilities for the functions f_1 and f_2 respectively. Let $p_{Q.1}^{1(n)}$ be the probability that at the end of n iterations, GA results in a population containing the best string from a population $Q \in \mathcal{Q}$ for the fitness function f_1 . Similarly, $p_{Q.1}^{2(n)}$ is that for f_2 . Then for any population $Q \in \mathcal{Q}$, $p_{Q.1}^1 \geq p_{Q.1}^2$, and in general $p_{Q.1}^{1(n)} \geq p_{Q.1}^{2(n)}$.

Proof:

Let Q be any population and Q' be a population containing one of the best strings. Then

$$\begin{aligned} p_{Q,1}^1 &= \sum_{Q' \in E_1} p_{Q,Q'}^1 \\ &= \sum_{Q' \in E_1, S_2 \notin Q'} p_{Q,Q'}^1 + \sum_{Q' \in E_1, S_2 \in Q'} p_{Q,Q'}^1 \\ &\geq \sum_{Q' \in E_1, S_2 \notin Q'} p_{Q,Q'}^1 \\ &\geq p_{Q,1}^2. \end{aligned}$$

Note that the equality holds when $S_2 \in Q$.

Now,

$$\begin{aligned} p_{Q,1}^{1(2)} &= \sum_{Q_1} p_{Q,Q_1}^1 p_{Q_1,1}^1 \\ &= \sum_{Q_1} p_{Q,Q_1}^1 \left(\sum_{Q' \in E_1, S_2 \notin Q'} p_{Q_1,Q'}^1 + \sum_{Q' \in E_1, S_2 \in Q'} p_{Q_1,Q'}^1 \right) \\ &\geq \sum_{Q_1} p_{Q,Q_1}^1 \left(\sum_{Q' \in E_1, S_2 \notin Q'} p_{Q_1,Q'}^1 \right) \\ &\geq p_{Q,1}^{2(2)}. \end{aligned}$$

Similarly, by using mathematical induction,

$$p_{Q,1}^{1(n)} \geq p_{Q,1}^{2(n)}. \spadesuit$$

□

It is now clear from the theorem that the probabilities in reaching to a population containing one of the best strings will be higher if the number of strings having the highest fitness function value is more. That is, the ϵ -optimal stopping time for the function f_2 is also an ϵ -optimal stopping time for f_1 . Thus, in this article, we shall deal with the functions which possess exactly one optimum string. These fitness functions are termed as single optimal fitness functions. Let $W = \{fit : fit \text{ is a fitness function and it has exactly one optimal string} \}$.

Definition 5. A fitness function fit is said to be a single optimal fitness function if $fit \in W$.

Another important problem is related to the distribution of the fitness values over the search space (set of 2^L strings). The inherent assumption in the implementation of GAs is that the variations in the fitness function values are less if the Hamming distance between the strings is less. But this assumption is not always true for certain fitness functions (e.g., the minimum deceptive problem [10]). For convenience, let us demonstrate a function in which the above assumption is not true.

Example 1 : Let $S^* = 111111 \dots 111$ and

$$\begin{aligned} fit(S^*) &= L + 1 \\ fit(S) &= D(S, S^*) ; S \neq S^* . \end{aligned}$$

Where the Hamming distance $D(S, S^*)$ between S and S^* is the number of bits at which the alleles are different. Note that, fit takes only $L+1$ values. Once the operators of GAs generates a population whose fitness value is $fit(000000 \dots 000)$ high mutation probability will probably help the process to obtain the global optimal solution. Section 4.4 discusses the different strategies in selecting the mutation probabilities.

4.3. Upper bound for optimal stopping time

In this subsection an upper bound for ϵ -optimal stopping time is obtained for an EGA. The fitness function under consideration is a single optimal fitness function. Let the fitness function fit have only one optimal string $S^* \in \mathcal{S}$, i.e.,

$$fit(S^*) > fit(S), \quad \forall S \in \mathcal{S}; \quad S \neq S^* .$$

Consider the population Q' consisting of M copies of a string $\overline{S^*}$, the complement of S^* where S^* is the optimal string, i.e., $D(S^*, \overline{S^*}) = L$.

Theorem 3. *Let $s_{Q,1}$ represent the probability of reaching a population containing S^* from Q in one iteration with mutation operation alone. Then $s_{Q,1} \geq s_{Q',1} \forall Q$.*

Proof:

It follows from the principles of probability theory. □

Theorem 4. *Let $p_{Q,1}$ be the probability of reaching a population containing the optimal string from Q in one iteration using selection, crossover & mutation operations.*

Then $p_{Q,1} \geq p_{Q',1}$ for all Q .

Proof:

Let $s_{Q,1}$ be the probability of reaching a population containing the optimal string from Q by using mutation operation alone. Let $Q = \{S_1, \dots, S_M\}$. be the given population.

Let ρ_i represent the probability of selecting the string S_i to the mating pool. Then $0 < \rho_i < 1$ & $\sum_{i=1}^M \rho_i = 1$.

Let after selection, the possible no. of mating pools be τ and they may be represented by Q_1, Q_2, \dots, Q_τ .

Note that Probability (obtaining a population $Q_i = \{S_{i1}, \dots, S_{iM}\}$ where $S_{ij} \in Q = \forall_{i,j}$)

$$= \prod_{j=1}^M \rho_{ij}. \quad \forall i = 1, 2, \dots, \tau$$

$$\text{Then } \sum_{i=1}^{\tau} \prod_{j=1}^M \rho_{ij} = 1.$$

Given a mating pool $Q_i; 1 \leq i \leq \tau$, the possible no. of populations that may be obtained after crossover be V_i and they may be represented by $Q_{i1}, Q_{i2}, \dots, Q_{iV_i}$. Let the probability of obtaining the population Q_{ij} from Q_i by using the crossover population alone be represented by W_{ij} .

Where $0 < W_{ij} < 1 \quad \forall j = 1, 2, \dots, V_i$ and $i = 1, 2, \dots, \tau$ & $\sum_{j=1}^{V_i} W_{ij} = 1 \quad \forall i = 1, 2, \dots, \tau$.

Then the possible populations that may be obtained after selection & crossover are $Q_{ij}; j = 1, 2, \dots, V_i; i = 1, 2, \dots, \tau$ with probabilities $W_{ij} \prod_{k=1}^M \rho_{ik}$.

Note that $\sum_{i=1}^{\tau} \sum_{j=1}^{V_i} W_{ij} \prod_{k=1}^M \rho_{ik} = 1$

Then

$$\begin{aligned} p_{Q.1} &= \sum_{i=1}^{\tau} \sum_{j=1}^{V_i} W_{ij} \prod_{k=1}^M \rho_{ik} s_{Q_{ij}.1} \\ &\geq \sum_i \sum_j W_{ij} \prod_{k=1}^M \rho_{ik} s_{Q'.1} \\ &= s_{Q'.1} \sum_{i=1}^{\tau} \sum_{j=1}^{V_i} W_{ij} \prod_{k=1}^M \rho_{ik} \\ &= s_{Q'.1}. \end{aligned} \tag{9}$$

Thus $p_{Q.1} \geq s_{Q'.1}$.

Now, note that $p_{Q'.1} = s_{Q'.1}$ since

- i) The mating pool, formed after selection operation is performed on Q' , is nothing but Q' and
- ii) The population to be obtained after crossover operation on Q' is nothing but Q' .

□

Hence $p_{Q.1} \geq p_{Q'.1} \quad \forall Q$.

Now for Q' , $p_{Q'.1}$ can be obtained by computing the probability of mutating each bit position of at least one string of Q' .

$$\begin{aligned} p_{Q'.1} &= \binom{M}{1} (q^L) - \binom{M}{2} (q^L)^2 + \dots + (-1)^M \binom{M}{M} (q^L)^M \\ &= \sum_{r=1}^M \binom{M}{r} (q^L)^r (-1)^{r+1} \\ &= 1 - (1 - q^L)^M. \end{aligned} \tag{10}$$

Note: The minimum probability of reaching a population containing the best string is independent of the characteristics of the fitness function values. Since one can always assume without loss of generality that there exists the complement of S^* . It is also to be mentioned here that the expression for $p_{Q'.1}$ is independent of the probability of crossover p and the selection procedure.

2. Recall δ , defined in the proof of theorem 1.

$$\begin{aligned} \delta &= \text{Max}_Q (1 - p_{Q.1}) \\ &= 1 - \text{Min}_Q p_{Q.1} \\ &= 1 - p_{Q'.1} (\text{From Theorem 4}) \end{aligned}$$

Theorem 5. Let fit be a fitness function with exactly one optimal string S^* . Let $\overline{S^*}$ be the complement string of S^* and Q' be a population consisting of M copies of $\overline{S^*}$ where M is the population size. Let Q_0 be any population of size M . Then

$$p_{Q_0,1}^{(n)} \geq 1 - (1 - q^L)^{Mn} \quad (11)$$

Proof:

We have $p_{Q',1} = 1 - (1 - q^L)^M$. Then, from the above note (2) and the theorem (1), this theorem follows. ♠ □

Upper bound for ϵ -optimal stopping time : It follows from the above theorem that

$$\begin{aligned} p_{Q_0,1}^{(n)} &\geq 1 - (1 - q^L)^{Mn} && \text{for } Q_0 \notin E_1 \\ &= 1 && \text{for } Q_0 \in E_1 \end{aligned} \quad (12)$$

Thus, $p_{Q,1}^{(n)} \geq 1 - (1 - q^L)^{Mn} \quad \forall Q$. Let $0 < \epsilon < 1$. Now if $1 - (1 - q^L)^{Mn} \geq 1 - \epsilon$ for $n \geq N$ then N is an ϵ -optimal stopping time for the GA.

Note that,

$$\begin{aligned} 1 - \epsilon &\leq 1 - (1 - q^L)^{Mn}, \\ \iff \epsilon &\geq (1 - q^L)^{Mn}, \\ \iff n &\geq \frac{\log \frac{1}{\epsilon}}{M \log \frac{1}{1 - q^L}} = \frac{\log \epsilon}{M \log(1 - q^L)}. \end{aligned} \quad (13)$$

Let $N(\epsilon, M, q, L) = \frac{\log \epsilon}{M \log(1 - q^L)}$.

Note that given ϵ , M , q and L , $N(\epsilon, M, q, L)$ is an upper bound for ϵ -optimal stopping time by construction. Note that we do not have the knowledge of minimal ϵ -optimal stopping time for GAs. Hence, in the subsequent analysis we assume that $N(\epsilon, M, q, L)$ iterations will be performed during the process.

Remarks :

1. For a given ϵ , M , q and L , $N(\epsilon, M, q, L)$ is independent of the characteristics of the fitness function, crossover probability and the selection procedure.
2. Note that, if the starting population is not Q' then also ϵ -optimal string will be obtained in $N(\epsilon, M, q, L)$ iterations.
3. Given ϵ , q and L , $N(\epsilon, M, q, L)M$ (the product of $N(\epsilon, M, q, L)$ and M) is a constant. Note that, $N(\epsilon, M, q, L)M$ provides the number of strings searched upto $N(\epsilon, M, q, L)$ iterations. This means that the number of strings to be searched to obtain ϵ -optimal string is independent of the population size M .
4. It can be seen that for given M , q and L ,

$$\epsilon_1 < \epsilon_2 \implies N(\epsilon_1, M, q, L) > N(\epsilon_2, M, q, L).$$

It implies that the number of iterations required is more to obtain a more accurate solution.

5. It is also clear that, for given ϵ , q and L ,

$$L_1 > L_2 \implies N(\epsilon, M, q, L_1) > N(\epsilon, M, q, L_2).$$

This also coincides with our intuition that for a fixed ϵ if the length of the string increases the required number of iterations also increases.

6. Note that the expression for $N(\epsilon, M, q, L)$ has been derived from the following inequalities.

$$p_{Q'.1}^{(n)} \geq p_{Q'.1}^{(n)} \geq 1 - (1 - p_{Q'.1})^n .$$

$1 - (1 - p_{Q'.1})^n$ is taken to be $\geq 1 - \epsilon$. Note that Q' is a pessimistic choice of an initial population and the stopping time has been derived from the above inequality. Thus the stopping time derived is also a pessimistic one and we shall denote it by $N_{pes}(\epsilon, M, q, L)$.

So far we have discussed the stopping time keeping the mutation probability q as fixed. In the next section, we shall study the behavior of ϵ -optimal stopping time with different mutation probabilities.

5. Mutation Probability and ϵ -optimal Stopping Time

In the previous section, the behavior of $N(\epsilon, M, q, L)$ has been studied keeping the mutation probability q as constant. Here first of all we shall study the behavior of $N(\epsilon, M, q, L)$ if different values for q are considered. Then, we shall try to find the modified expression for ϵ -optimal stopping time with the assumption that the characteristics of the fitness function is well related to the hamming distance of the strings with the best string S^* . That is, the differences in the fitness function is such that they decrease with the reduction in $D(S, S^*)$, the hamming distance between S and the optimal string S^* .

$N_{pes}(\epsilon, M, q, L)$ represents an upper bound for ϵ -optimal time. Here we shall try to find the optimal value of q which will minimize $N_{pes}(\epsilon, M, q, L)$. A closer look at equation 13 reveals that for $q_1 < q_2 \leq 0.5$, $N_{pes}(\epsilon, M, q_1, L) > N_{pes}(\epsilon, M, q_2, L)$.

It follows that the optimal value of q for $N_{pes}(\epsilon, M, q, L)$ is 0.5. Note that 0.5 is a very high value for q in the implementation of a GA.

It has been found above that $N_{pes}(\epsilon, M, q, L)$ is minimum when $q = 0.5$. $q = 0.5$ provides the usual Monte Carlo search (pure random search) for optimal. Note that the mutation probability q is indeed very high ($q = 0.5$) for making $N_{pes}(\epsilon, M, q, L)$ minimum. In practice, the researchers take q to be very low in order to ape the usual genetic systems. Thus q being very high would go against the practice of the usual genetic system. But one can still find $q = 0.5$ to be useful provided the number of strings searched upto $N_{pes}(\epsilon, M, q, L)$ iterations is less than 2^L . We shall show below that the number of strings searched upto $N_{pes}(\epsilon, M, q, L)$ iterations is greater than 2^L .

Note that,

$$M.N_{pes}(\epsilon, M, q, L) = \frac{\log \frac{1}{\epsilon}}{\log \frac{1}{1-0.5^L}} . \quad (14)$$

Now

Substituting MN with 2^L , we get

$$\frac{\log \frac{1}{\epsilon}}{\log \frac{1}{1-0.5^L}} \leq 2^L \quad (15)$$

$$\begin{aligned} \Leftrightarrow \log \frac{1}{\epsilon} &\leq 2^L \log \frac{1}{1-0.5^L}, \\ \Leftrightarrow \epsilon &\geq \left(1 - \frac{1}{2^L}\right)^{2^L}. \end{aligned} \quad (16)$$

Note that, as $L \rightarrow \infty$, $(1 - 0.5^L)^{2^L} \rightarrow e^{-1} \geq 0.37$. Which implies, that for $q = 0.5$ and for sufficiently large values of L , if the number of strings searched is 2^L then it can be stated that the fitness value obtained will be at least equal to $F_1(1 - 0.37) = 0.63F_1$.

Note that pessimistic stopping time always assumes that the starting population for any iteration is Q' . This is not the case in practice. Note also that for many fitness function, even if the initial population is Q' , it will result in some other population after a few iterations excepting for minimum deceptive problem. Observe that, usually, for minimum deceptive problem, $D(S, S^*)$ is less \Leftrightarrow difference between $fit(S)$ and $fit(S^*)$ is high. On the other hand, for the general optimization problems, $D(S, S^*)$ is less \Leftrightarrow difference between $fit(S)$ and $fit(S^*)$ is also less.

Thus from the above observation, we shall study the stopping time for EGA under the following assumption.

Assumption : Let S^* be the optimal string. Let there exist an integer d , $1 \leq d \leq L - 1$ such that for every two strings S_1 and S_2 , $S_1 \neq S_2$, $D(S_1, S^*) \leq d$, $D(S_2, S^*) > d$ implies $|fit(S_1) - fit(S^*)| < |fit(S_2) - fit(S^*)|$

Remarks :

1. The above assumption provides a cut off point d for the Hamming distance between the optimum string and the other strings. Note that, if the fitness function satisfies the above assumption for a specific d and if a string S is present in a starting population for an EGA such that $D(S, S^*) \leq d$ then the best string of any subsequent iteration will possess the Hamming distance $\leq d$ with the optimal one.
2. The above assumption holds good for many practical problems. It is not valid for minimum deceptive problems.
3. Note that d is taken to be a positive integer and it is strictly less than L . If there does not exist any d which satisfies the above assumption (or, $d = 0$ or $d = L$) then the fitness function is deceptive.
4. Suppose a fitness function fit satisfies the above assumption for any $d < d_1$. In that case, the fitness function will be regarded as —well related— with respect to the Hamming distance.

Let $Q \notin E_1$ be the population at the start of an iteration and S^* be the best string, and $D(S^*, Q) = d$. Then,

$$p_{Q.1} \geq 1 - [1 - q^d(1 - q)^{L-d}]^M,$$

and,

$$p_{Q,1}^{(n)} \geq 1 - [1 - q^d(1 - q)^{L-d}]^{Mn}.$$

Now, ϵ -optimal string will be obtained if

$$n \geq \frac{\log \frac{1}{\epsilon}}{M \log \frac{1}{1 - q^d(1 - q)^{L-d}}}, \quad \text{for } d \leq \frac{L}{2}. \quad (17)$$

Assuming L , M and ϵ as constants the minimum value of n is obtained when $q = \frac{d}{L}$.

It is also to be noted that the value for q does not exceed 0.5 and hence the optimal value for q is $\min(0.5, \frac{d}{L})$. Thus the number of iterations required to obtain an ϵ -optimal string is

$$n \geq \frac{\log \frac{1}{\epsilon}}{M \log \frac{1}{1 - (\frac{d}{L})^d (\frac{L-d}{L})^{L-d}}}, \quad \text{for } d \leq \frac{L}{2}. \quad (18)$$

This expression is obtained with an optimistic assumption and hence, we shall term it as an optimistic ϵ -optimal stopping time and denote it by N_{op} , i.e.,

$$N_{op}(\epsilon, M, d, L) = \frac{\log \frac{1}{\epsilon}}{M \log \frac{1}{1 - (\frac{d}{L})^d (\frac{L-d}{L})^{L-d}}} \quad (19)$$

Remarks :

1. Unlike N_{pes} , $N_{op}(\epsilon, M, d, L)$ is not independent of the characteristics of the fitness function for given ϵ , M , d and L . But it is independent of crossover probability and the selection procedure.
2. Given ϵ , d and L , $N_{op}M$ (the product of $N_{op}(\epsilon, M, d, L)$ and M) is a constant. Note that, $N_{op}(\epsilon, M, q, L)M$ provides the number of strings searched upto $N_{op}(\epsilon, M, q, L)$ iterations. This means that the number of strings to be searched to obtain the ϵ -optimal string in N_{op} many iterations is independent of the population size M .
3. It can be seen that for given M , d and L ,

$$\epsilon_1 < \epsilon_2 \implies N_{op}(\epsilon_1, M, d, L) > N_{op}(\epsilon_2, M, d, L).$$

It is also clear that, for given ϵ , d and M ,

$$L_1 > L_2 \implies N_{op}(\epsilon, M, d, L_1) > N_{op}(\epsilon, M, d, L_2).$$

4. Now, we are in a position to explain the affinity of researchers to assume q to be very small. Let the fitness function be well related to the Hamming distance between the best string and the best obtained so far. Since, the characteristics of the fitness function are completely unknown, the users will appreciate any improvement in the fitness function value due to even a single change in any bit position of the string. More specifically, the intention of GA users is to make a small change to improve the result in hand. Then to obtain a better

string compared to the string obtained so far, one needs to change $d \leq L/2$ (say) bit positions (i.e., reduce the Hamming distance by d) and assume mutation probability q to be $\frac{d}{L}$. For example, if a user expects an improvement by changing only 1 bit then the suitable value for q will be 0.01 for $L = 100$ and $d = 1$. q will be 0.001 for $L = 1000$ and $d = 1$ and so on.

5. It can also be seen that the number of strings searched to get an ϵ -optimal string in N_{op} iterations is $\leq 2^L$ for sufficiently large L and for any $\epsilon > 0$. To estimate the value of ϵ for $M.N \leq 2^L$, let us substitute MN by 2^L in (18) and we get

$$\begin{aligned} \log \frac{1}{\epsilon} &\leq 2^L \log \frac{1}{1 - (\frac{d}{L})^d (\frac{L-d}{L})^{L-d}}, \\ \text{or } \epsilon &\geq [1 - (\frac{d}{L})^d (\frac{L-d}{L})^{L-d}]^{2^L}. \end{aligned} \quad (20)$$

Let us consider $\theta = \frac{d}{L}$, then from (20) we have

$$\begin{aligned} \epsilon &\geq (1 - \theta^{\theta L} \cdot (1 - \theta)^{(1-\theta)L})^{2^L} \\ \text{or } (1 - \epsilon^{\frac{1}{2^L}})^{\frac{1}{L}} &\leq \theta^\theta \cdot (1 - \theta)^{1-\theta}. \end{aligned}$$

Note that $\theta^\theta \cdot (1 - \theta)^{1-\theta}$ is minimum when $\theta = 0.5$ and the minimum value of $\theta^\theta \cdot (1 - \theta)^{1-\theta}$ is 0.5. Thus if $\theta < 0.5$ then $\theta^\theta \cdot (1 - \theta)^{1-\theta} > 0.5$. Let us assume

$$0.5 < \theta^\theta \cdot (1 - \theta)^{1-\theta} < 0.5 + \rho, \text{ for } 0 < \rho < 0.5. \quad (21)$$

Then, note that $(1 - (0.5 + \rho)^L)^{2^L} \rightarrow 0$ as $L \rightarrow \infty$ and $\epsilon \geq (1 - (0.5 + \rho)^L)^{2^L}$.

For an appropriate value of q one can find an ϵ -optimal string (for any given ϵ) by searching $\leq 2^L$ strings for sufficiently large value of L . It is to be noted that if q decreases then ρ increases and consequently $(1 - (0.5 + \rho)^L)^{2^L}$ decreases and hence ϵ can be taken to be a small value. But it should be remembered that q being small implies that the fitness function is well behaved in the sense of convergence of GAs. That is, it has been shown theoretically that GAs are indeed useful in searching optimal solution and the number of strings to be searched is less than 2^L if (1) assumption 1 is satisfied and (2) the starting population contains a string whose Hamming distance with the optimal string is less than or equal to d . Investigations are to be conducted if assumption 1 does not hold. Mathematical treatment of stopping times, where the effects of selection and crossover are also taken into consideration, is necessary.

6. Conclusions and Discussion

In this paper, the problem of optimal stopping time for elitist model of genetic algorithms has been discussed. In this model, the best string of the previous iteration is preserved in the present population if the fitness function value of each offspring is worse than that of the previous best.

The maximization problem is considered here and took the mutation probability to be constant throughout the process. The binary representation scheme is considered in this article. An objective criterion for stopping the process in the implementation of GAs has been suggested. Consequently ϵ -optimal stopping times have been derived from pessimistic and optimistic points of view for the case of single optimal fitness function. For the pessimistic approach it has been found that the value of optimal mutation probability is 0.5.

It has also been shown that the number of strings to be searched to obtain ϵ -optimal string is less than 2^L from the optimistic point of view and for sufficiently large L . This fact ensures the utility of genetic algorithms in solving complex optimization problems. It has also been shown that the mutation probability is small for optimistic ϵ -optimal stopping time whereas it is 0.5 for pessimistic ϵ -optimal stopping time.

We have found the values of N_{pes} for $L = 80$ and $M = 100$. Now $2^L = 1208925819614629174706176$. For $\epsilon = 1/e$ and $q = 0.5$, the value of N_{pes} is 12089258196146291747062. For $q = 0.2$ and ϵ being the same as above, N_{pes} is found to be a 53 digit integer. For $q = 0.1$, N_{pes} is found to be an integer with minimum 70 digits. If ϵ is 0.2 and $q = 0.5$, the N_{pes} is 19456910474082525747527. It is found experimentally that as ϵ decreases, N_{pes} increases. As q decreases from 0.5, N_{pes} increases. As M increases, N_{pes} decreases.

Similarly, the values of N_{op} are found for $L = 80$ and $M = 50$. The value of N_{op} for $\epsilon = 0.1$ and $d = 20$ is 1587704615975069749. The value of N_{op} for $\epsilon = 0.1$ and $d = 10$ is 566998412210. For $\epsilon = 0.1$ and $d = 5$, $N_{op} = 6109419$. For $\epsilon = 0.1$ and $d = 1$, $N_{op} = 10$. N_{op} decreases as d decreases. Some more values of N_{op} are given below for $d = 1$ for the convenience of the readers when $L = 80$ and $M = 50$. For $\epsilon = 0.05$, $N_{op} = 13$. For $\epsilon = 0.005$, $N_{op} = 23$. For $\epsilon = 0.00001$, $N_{op} = 50$.

The values of N_{pes} and N_{op} provide combinatorial verification of the theoretical results stated in the article. It can be seen from the above values that $N_{op}/2^L$ is small even for $d = 20$ when $\epsilon = 0.1$, $L = 80$ and $M = 50$. When $d = 1$, the values of N_{op} are extremely small.

A detailed theoretical investigation should be made on the selection process and the utility of crossover operations. The optimistic approach reflects that a varying mutation probability with respect to iteration number may be useful in faster convergence of the process. It is also to be noted that the similar study can be made to derive the ϵ -optimal stopping time for general genetic algorithms where the best string is preserved in a separate location outside the population.

As far as our knowledge goes, stopping times for GAs, have not been derived mathematically in the literature. This article is an attempt in this regard. The results obtained may not be as impressive as they should be. Further mathematical investigations are therefore necessary to provide more general and realistic versions of stopping times for GAs. Investigations are also

necessary to judge theoretically the effect of selection and crossover operators on the stopping times, though the obtained stopping times are valid for an EGA with selection, crossover and mutation operations.

Acknowledgement : The work was carried out while Prof. S. K. Pal held Jawaharlal Nehru Fellowship.

References

- [1] S. K. Pal and P. P Wang (Eds.) *Genetic Algorithms for Pattern Recognition*, CRC Press, Boca Raton, USA, June 1996.
- [2] S. Bandyopadhyay, C. A. Murthy and S. K. Pal, "Pattern classification using Genetic Algorithms", *Pattern Recognition Letters*, vol. 16, pp. 801-808, 1995.
- [3] S. Bandyopadhyay and S. K. Pal, "Pattern classification with Genetic Algorithms : Incorporation of chromosome differentiation", *Pattern Recognition Letters*, vol. 18, pp. 119-131, 1997.
- [4] R. Belew and L. Booker (eds.), *Proceedings of the fourth int. conf. on genetic algorithms*. Los Altos, CA: Morgan Kaufmann Publishers, 1991.
- [5] D. Bhandari, C. A. Murthy, and S. K. Pal, "Genetic algorithm with elitist model and its convergence," *Int. J. of Pattern Recognition and Artificial Intelligence*, vol. 10, pp. 731-747, 1996.
- [6] D. Bhandari, N. R. Pal, and S. K. Pal, "Directed mutation in genetic algorithms," *Information Sciences*, vol. 79, pp. 251-270, 1994.
- [7] S. Bornholdt and D. Graudenz, "General asymptotic and neural networks and structure design by genetic algorithms," *Neural Networks*, vol. 5, pp. 327-334, 1992.
- [8] R. Das and D. Whitley, *The only challenging problems are deceptive : global search by solving order - 1 hyperplane*, in [4], pp. 166-173.
- [9] L. Davis (eds), *Genetic Algorithms and Simulated Annealing*. London: Pitman Publishing, 1987.
- [10] T. E. Davis and C. J. Principe, *A simulated annealing like convergence theory for the simple genetic algorithm*, in [4], pp. 174-181.
- [11] W. Feller, *An Introduction to Probability Theory and its Applications (Vol. I)*. New Delhi: Wiley Eastern Pvt. Ltd., 1972.
- [12] D. E. Goldberg, *Genetic Algorithms : Search, Optimization and Machine Learning*. : Addison Wesley Publishing Company, Inc., 1989.
- [13] M. Hulin, *Analysis of schema distribution*, in [4], pp. 190-196.
- [14] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*. : Springer Verlag, 1992.

- [15] A. E. Nix and M. D. Vose, "Modeling genetic algorithms with Markov chains," *Annals of Mathematics and Artificial Intelligence*, vol. 5, pp. 79–88, 1992.
- [16] S. K. Pal, D. Bhandari, and M. K. Kundu, "Genetic algorithms for optimal image enhancement," *Pattern Recognition Letters*, vol. 15, no. 3, pp. 261–271, 1994.
- [17] S. K. Pal and D. Bhandari, "Genetic algorithms with fuzzy fitness function using cellular neural networks for object extraction," *Fuzzy Sets and Systems*, vol. 65, no. 2–3, pp. 129–139, 1994.
- [18] S. K. Pal and D. Bhandari, "Selection of optimal set of weights in a layered network using genetic algorithms," *Information Sciences*, vol. 80, no. 3–4, pp. 213–234, 1994.
- [19] N. J. Radcliffe, *Forma analysis and random respectful recombination*, in [4], pp. 222–229.
- [20] G. Rudolph, "Convergence analysis of canonical genetic algorithms", *IEEE Tran. on Neural Networks*, vol. 5, no. 1, pp. 96–101, 1994.
- [21] A. N. Shiriyayev, *Optimal stopping rules*. New York : Springer-Verlag, 1977.
- [22] W. Siedlecki and I. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, pp. 335–347, 1989.
- [23] M. D. Vose, "Generalizing the notion of schema in genetic algorithms," *Artificial Intelligence*, vol. 50, pp. 385–396, 1988.
- [24] M. D. Vose, "Modeling simple genetic algorithms," in *Foundations of genetic algorithms II*, D. Whitley Ed., Morgan Kauffmann, 1993.
- [25] M. D. Vose and G. E. Liepins, "Punctuated equilibria in genetic search," *Complex Systems*, vol. 5, pp. 31–44, 1991.
- [26] D. Whitley, T. Starkweather, and C. Bogart, "Genetic algorithms and neural networks : optimizing connections and connectivity," *Parallel Computing*, vol. 14, pp. 347–361, 1990.
- [27] J. Suzuki, "A markov chain analysis on a genetic algorithm", *Proc. of the fifth international conference on genetic algorithms*, pp. 146-153, Morgan Kaufmann Publishers, 1993.
- [28] J. Suzuki, "A markov chain analysis on simple genetic algorithm", *IEEE Transactions on Systems, Man and, Cybernetics*, vol. 25, no. 4, pp. 655–659, 1995.
- [29] K. A. Dejong, *An analysis of the behaviour of a class of genetic adaptive systems*, Ph. D. thesis, Department of Computer and Communication Science, Univ. of Michigan, Ann Arbor, 1975.